LANGUAGE DOCUMENTATION & CONSERVATION

Vol. 6 (2012), pp. 104-134
http://nflrc.hawaii.edu/ldc/
http://hdl.handle.net/10125/4500

Documenting Endangered Languages with Linguist's Assistant

Stephen Beale
University of Maryland, Baltimore County

The Linguist's Assistant (LA) is a practical computational paradigm for describing languages. LA contains a meaning-based elicitation corpus that is the starting point and organizing principle from which a linguist describes the linguistic surface forms of a language using LA's visual lexicon and grammatical rule development interface. This paper presents a brief overview of the semantic representation system that we have designed and discusses the meaning-based elicitation methodology. Next it describes the process by which the linguist enters lexical and grammatical information, then it discusses the ancillary functions of LA that allow for an efficient and accurate language description as well as the facilities for producing written documentation of the language. Videos are included to demonstrate the major functionality of LA.[1]

1. INTRODUCTION. The Linguist's Assistant (LA)[2] is a practical computational paradigm for describing languages. LA approaches the complex task of language description from two directions. From one side, LA is built on a comprehensive semantic foundation. We combine a conceptual, ontological framework with detailed semantic features that cover (or is a beginning towards the goal of covering) the range of human communication. An elicitation procedure has been built up around this central, semantic core that systematically guides the linguist through the language description process, during which the linguist builds a grammar and lexicon that 'describes' how to generate target language text from the semantic representations of the elicitation corpus. The result is a meaning-based 'how to' guide for the language: how does one encode given semantic representations in the language?

Coming at the problem from the other side, LA also allows the linguist to collect language data in a more conventional manner—from naturally occurring corpora that are analyzed in a form-based study. For example, a linguist in Vanuatu might want to explore alienable vs. inalienable possession (Beale & Allman 2011) and/or serial verb constructions. Naturally occurring texts that exemplify phenomena of interest such as these are, in the LA paradigm, first semantically analyzed using a convenient semi-automatic mark-up interface, in effect adding them to the standard elicitation corpus. Existing grammar rules and lexical information can then either be confirmed or adjusted, or new descriptive knowledge added that allows the built-in text generator to produce target text that is substantially

[1] Video examples used in this article are stored in ScholarSpace (http://hdl.handle.net/10125/4500). Playable versions in YouTube are linked to in the pdf version of this article and require an internet connection and browser to be viewed.

[2] I gratefully acknowledge the collaboration of Tod Allman, University of Texas, Arlington. Dr. Allman wrote and co-designed the LA, TA, and TBTA programs. All errors, inconsistencies, and unintelligible descriptions in this document are my own. Dr. Allman's main research interests lean more toward translation than language documentation, explaining his absence from this paper.

E-ISSN 1934-5275

equivalent to the elicited examples. The result is a form-based 'how did' guide for the language: how did a native speaker encode natural text?

We believe that the combination of semantically-motivated and form-motivated elicitation, analysis, and description provides an ideal balance (cf. Ameka et al. 2006:14ff). The meaning-based elicitation corpus is general and language-independent. It provides an efficient and relatively comprehensive standard for describing many different linguistic phenomena in a language. We have found it to be an invaluable starting point and agree with Ameka et al. that "despite the difficulties in executing this ideal... we believe this [meaning-based grammar] should remain an important descriptive goal." It is, however, impossible to produce a general meaning-based elicitation scheme that is not overly burdensome on the user. In addition, linguists typically know the 'interesting,' atypical, or difficult aspects of a language. This is where focused form-based analysis is invaluable.

A third approach to language description is encouraged in the LA framework: acquiring knowledge (lexicon and grammar) to cover pre-authored texts ('authored' in our context means that a semantic representation has been prepared). The semantically- and linguistically-motivated elicitations from the first two approaches above provide a solid foundation for lexicon and grammar development, but we have found that adding to that the experience and discipline of acquiring the knowledge necessary to generate actual texts is invaluable. This is usually the best opportunity for documenting phenomena that are more lexically dependent, since the vocabulary in the elicitation stages is quite limited. For this reason we include several pre-authored community development texts with LA.

Underlying all these approaches to knowledge acquisition in LA is a visual, semi-automatic interface for recording grammatical rules and lexical information. Grammatical rules in LA describe how a given semantic structure is realized in the language. The whole gamut of linguistic phenomena is covered, from morphological alternations to case frame specifications to phrase structure ordering to lexical collocations—and many others. These grammatical rules interplay with a rich lexical description interface that allows for assignment of word-level features and the description of lexical forms associated with individual roots. Currently, the linguist is responsible for the creation of rules, albeit with a natural, visual interface that is often able to set up the requisite input semantic structures automatically. We continue work on a module that will allow the semi-automatic generation of rules similar to research in the BOAS (McShane et al. 2002), the LinGO Grammar Matrix (Bender et al. 2010), PAWS (Black & Black 2009) and Avenue (Probst et al. 2003) projects. Such a module will, we believe, make LA accessible to a larger pool of linguists. We also provide a growing list of rule templates that linguists can use to describe common linguistic phenomena. Although multi-lingual re-use of grammatical descriptions along the lines of KPML (Bateman, 1997) was not a design goal in LA, we intend to describe in future publications how the KPML development cycle of copy (language resources from a somewhat related language), revise (using point-and-click interface tools) and validate (using test suites) can be achieved using LA.

Integrated with these elicitation and description tools is a text generator that allows for immediate confirmation of the validity of grammatical rules and lexical information.[3] We also provide an interface for tracking the scope and examples of grammatical rules.

[3] We do not currently plan on developing a parser that could utilize these resources.

This minimizes the possibility of conflicting or duplicate rules while providing the linguist a convenient index into the work already accomplished. And finally, we provide a utility for producing a written description of the language. After all, a computational description of a language is of no practical use (outside of translation applications) unless it can be conveniently referenced. LA not only provides a structured framework for elicitation and linguistic description, but, with further development and feedback from the linguistic community, it could also enable and eventually deliver a partial standard for the description of a language.

We target this paper to an audience composed mostly of field linguists who might be interested in using LA. The purpose of the paper is to introduce LA to this group, giving enough information so that a linguist will clearly understand the scope of the LA program, appreciate its possible use in a language description project, and acquire a sense for some of the practical details of its use. As such, we have avoided lengthy digressions of a more theoretical nature, both because of space constraints, and because such discussions might detract from the overall goal of the paper. We have, however, included references to other papers in which the theoretical basis of LA is discussed in detail. We direct the reader to Allman (2010) for the most comprehensive description of LA at a theoretical level.

Of course, LA is a work in progress. The underlying semantic formalism has been developed for multi-lingual use, but obviously such work is never complete; we continue to research issues in semantic representation and are open to revision based on the requirements of specific languages. We also continue research in the area of elicitation strategies. Obviously, the more diagnostic a set of elicitation examples is, the better and more efficient the resulting description. This competes with requirements for broad coverage and, importantly, ease of use. And as already stated, we are also actively researching the use of a semi-automatic rule discovery module (in the Avenue project tradition), which would further impact elicitation requirements. We will also be integrating strategies from typology questionnaire approaches (see BOAS, the LinGO Grammar Matrix and PAWS). And finally, we require feedback and further specifications from the linguistic community regarding the written description that LA produces for a language.

LA has been used to produce extensive grammars and lexicons for Jula (a Niger-Congo language), Kewa (Papua New Guinea), North Tanna (Vanuatu), Korean, and English; see Allman & Beale (2004, 2006) for details. Work continues in two languages of Vanuatu, with additional languages planned in the near future. The resulting computational resources have been used in our separate document authoring and translation application to produce a significant amount of high-quality translations in each of these languages. Beale et al. (2005) and Allman & Beale (2004, 2006) give more information on using LA in translation projects, and for documentation on the evaluations of the translations produced.[4] We argue that the high-quality results achieved in translation projects demonstrates the quality and coverage of the underlying language description that LA produces. As will be described below, LA is appropriate for use in both analytic and synthetic languages, and in the case of the latter, both agglutinative and fusional languages.

[4] LA can be used as part of this larger, translation-oriented program called TA (The Translator's Assistant, for translating health and community development materials, as well as the ability to 'author' new texts) or TBTA (The Bible Translator's Assistant, for those interested in Bible translation).

LA is available for academic research and non-profit applications. Tutorials and related papers are also available, although a significant portion of our planned work is to produce better tutorials and workshop materials. We are eager to assist any computationally-minded linguists who are interested in using LA. We also continue work on using LA to describe languages, and we plan on reporting these additional results in the near future. LA runs on Windows XP or later. It has also been used successfully on Macs running the Parallels Windows environment.

We emphasize that LA is a work in progress. In any practical product with complex theoretical underpinnings, there is a development loop where a 'critical mass' of theory is implemented, the surrounding support tools are created and tested, the product is used and evaluated, and then work begins on improving the theoretical base. LA is somewhere in the late stages of the 'being used and evaluated' step of this cycle. We certainly intend to improve the theoretical basis of each aspect of the product as time goes on, in large part as a result of the feedback, suggestions, and criticism of our users.

▶ Play Demo in YouTube

VIDEO 1. An overview of the Linguists Assistant methodology.

2. SEMANTIC REPRESENTATION IN LA. So, why exactly do we need a semantic representation in a language description system? At a very high level of description, LA seeks to specify in semantic representations a subset of possible written communication. These semantic representations then become the starting point and organizing principle from which a linguist describes the linguistic surface forms used in a language. Obviously we will not completely succeed in corralling all of human communication. For one thing, there are many more concepts (especially objects) in the world than any one linguist would care to elicit. But we do aspire to include a set of concepts that will allow a linguist in any language to adequately describe a wide variety of linguistic features of the language. More importantly, we claim that our set of semantic features, which describe phenomena such as temporal relationships, aspect, mood, reference, spatial orientation, and so on, are sufficient to describe the great majority of non-conceptual meaning in human communication.

Allman (2010) provide detailed descriptions of the LA semantic representation system. For the purposes of this paper, we present below several videos as overviews for those readers who are interested.

2.1. THE VISUAL PRESENTATION OF SEMANTIC REPRESENTATIONS. In this section a series of videos will be used to illustrate how LA deals with semantic representations.

▶ Play Demo in YouTube

VIDEO 2. An overview of the semantic representation system created for and used in LA.

2.2. ONTOLOGY.

▶ Play Demo in YouTube

VIDEO 3: A discussion of the conceptual framework used in LA.

2.3. SEMANTIC FEATURES.

▶ Play Demo in YouTube

VIDEO 4: A description of the semantic features that are used to modify the basic conceptual information.

3. MEANING-BASED ELICITATION AND OVERALL APPROACH TO LANGUAGE DESCRIPTION.

3.1. LINGUISTIC BACKGROUND AND RELATED RESEARCH IN COMPUTATIONAL ELICITATION TECHNIQUES. We view ourselves to be in the tradition of *Studying and Describing Unwritten Languages* (Bouquiaux & Thomas 1992) and Comrie & Smith's (1977) linguistic questionnaire. Our goal is to make these kinds of resources into a computational tool that becomes the organizing principle behind language description. Of course the central contribution of our work is the inclusion of grammatical analysis and a built-in text generation system that provides immediate feedback.

Linguistically speaking we have been most influenced by the work of Payne (1997) and Givón (1984). Our aim was to make LA a descriptive framework, not tied to a particular theory such as HPSG (cf. the LinGO Grammar Matrix below) or PATR (cf. PAWS below). In the world of Computational Linguistics we descend from the semantic approach to machine translation at Carnegie Mellon University in the 1990s (Nirenburg & Raskin 2004; Nyberg & Mitamura 1992). Our visual grammatical rule interface, described in section 4, Encoding Lexical and Grammatical Knowledge, was designed to be powerful but user-friendly, enabling the user to move from the underlying semantic representations of the elicitation corpus to restructured structures closer to the underlying target language to the actual surface structures themselves. At heart we are practical, descriptive computational linguists. The developers were linguistically trained, in part, by the Summer Institute of Linguistics (SIL); the author performed linguistic surveys for SIL in Northeast Asia and extended linguistic fieldwork in Vanuatu. We should note that care has been taken to make LA a general linguistic tool that is not at all tied to the goals of SIL. We are also responsible for other related translation-oriented tools, at times known as TA (Translator's Assistant) and TBTA (The Bible Translator's Assistant). These 'document authoring' systems are aimed at the semantic authoring and translation of community development literature (TA) and the Bible (TBTA). Such work is relevant to LA only by virtue of the fact that thousands of lines of texts have been authored and represented in our semantic representations, as well as successfully translated into a wide variety of language families, all lending credence to our underlying descriptive approach.

LA is somewhat related to a line of computational research that we would categorize as grammatical typology questionnaires. BOAS (McShane et al. 2002), the LinGO Grammar Matrix (Bender et al. 2010) and PAWS (Black & Black 2009) all fit into this paradigm. The author continues to participate in ongoing research at the University of Maryland that is descended from the BOAS system. All these systems extract salient properties of a language through typological questionnaires and then produce computational resources of varying utility. We applaud this work, and as mentioned, are actively involved in it. We intend to integrate this approach into LA for certain phenomena. However, the typology question-

naire approach is limited to creating approximate grammars; Bender et al. (2010) describe the LinGO Grammar Matrix as a 'rapid prototyping' tool. Furthermore, the phenomena that typology questionnaire approaches can cover are also easily captured using LA. Writing rules to represent phrase structure word ordering and phenomena such as case, agreement, nominal declensions and the like is more or less run-of-the-mill for us; in section 4 we describe types of rules that can implement each. We readily acknowledge, however, that different phenomena can be most effectively described using different techniques, so we are eager to continue pursuing the typology questionnaire paradigm. It is clear, though, that LA goes beyond the goals of typology questionnaires. Our meaning-based questionnaire aims at eliciting linguistic information that would be difficult or impractical to acquire through typological questioning. The computational grammar and lexicon produced in an LA-type language description project are meant to be comprehensive and complete insofar as they will be able to be used in our integrated text generator to produce accurate translations of the entire semantic elicitation corpus. Furthermore, with additional work that is mainly focused on adding vocabulary, a large corpus of semantically authored texts (including newly authored documents relevant to a particular community) can be accurately translated.

Another computational approach to linguistic description that is related to our own is the Avenue Project at Carnegie Mellon University (Probst et al. 2003). The Avenue project is a machine translation system oriented towards low-density languages. It consists of two central parts: 1) the pre-run-time module that handles the elicitation of data and the subsequent automatic creation of transfer rules, and 2) the actual translation engine. The former module is the only one relevant to LA:

> "The purpose of the elicitation system is to collect a high-quality, word-aligned parallel corpus. Because a human linguist may not be available to supervise the elicitation, a user interface presents sentences to the informants. The informants must be bilingual and fluent in the language of elicitation and the language being elicited, but do not need to have training in linguistics or computational linguistics. They translate phrases and sentences from the elicitation language into their language and specify word alignments graphically.
>
> The rule-learning system takes the elicited, word-aligned data as input. Based on this information, it infers syntactic transfer rules…. The system also learns the composition of simpler rules into more complicated rules, thus reducing their complexity and capturing the compositional makeup of a language (e.g., NP rules can be plugged into sentence-level rules). The output of the rule-learning system is a set of transfer rules that then serve as a transfer grammar in the run-time system." (Probst et al. 2003:247–248)

At a high level, this is very close to our approach. However, the Avenue system was never targeted for use by field linguists. LA differs from Avenue in several important features, notably our underlying semantic representation, as opposed to Avenue's transfer (source surface language to target surface language) approach. LA attains a greater practicality than Avenue because we insert a human into the rule-creation process, albeit in a situation in which that human can rely heavily on the visual (and semi-automatic) rule creation interface that will be described below.

3.2 OVERVIEW OF LA LANGUAGE ACQUISITION. What is our basic approach? There is nothing magical about our semantic questionnaire. It is simply an attempt to organize and present a wide variety of semantic possibilities that are present in human communication. It was developed in a straightforward manner: by going through our semantic representation system and creating sets of 'minimal pair' (or triples, etc.) inputs that will allow the linguist to describe the target language realizations for each semantic feature and for each type of conceptual argument structure. We sometimes refer to the questionnaire as a 'Grammar Introduction' module. It currently consists of a corpus of approximately 300 sentences. Each of the sentences illustrates a particular semantic concept, feature, or construction/argument structure. After working through the Grammar Introduction, the resulting computational model of the language should describe, for example, phrase structure ordering, case roles, agreement, a variety of verbal aspects and moods, pronoun systems, relative clauses formed on a variety of semantic roles, patient propositions (object complements) formed with a variety of matrix events, different types of adverbial clauses, different types of questions, and so on.

Each sentence in the Grammar Introduction has a semantic representation that was manually created by the developers. Each sentence also has an English translation and, when helpful, some explanatory text about what that sentence is seeking to illustrate. The key benefits of having the semantic representation for each sentence are:

- It makes explicit the concepts, features and structures that are of interest for each sentence.

- It typically sets up the input structure for the rules that need to be written.

- After the user has written rules to describe the phenomena inherent in the semantic representation, the integrated text generator automatically generates a translation of the underlying semantics. This gives immediate feedback. A loop of rule writing, immediate feedback followed by rule revision is ideal for the analysis and description of linguistic phenomena.

- The semantic representation, its stored English translation and explanatory text, all applicable rules, and the resulting automatically generated target language text all become the basis for documenting the linguistic realization of the phenomena being addressed in the rule and/or input semantic representation.

The overall, project-level steps taken to describe a language can then be summarized as:

1. Elicit translations for the sentences and texts in the questionnaire.

2. Manually analyze these translations.

3. For each sentence in the questionnaire:
- Write rules using the visual rule interface (we will expand on this step below).

• Immediately test the rules by automatically producing translations of the underlying semantic representations and comparing these to the manually elicited translations.
• Revise the rule as necessary and retest.

4. Extend the meaning-based elicitation with form-based elicitations and naturally occurring texts that focus on specific target language phenomena.

5. Optionally extend (or intersperse) the elicitation-based acquisition with acquiring necessary lexical and grammatical knowledge to translate actual pre-authored texts.

To demonstrate the utility of the Grammar Introduction module, Figure 1 shows the number of rules that were written for the Grammar Introduction module in Kewa, a clause chaining language with a switch reference system spoken in Papua New Guinea.[5] Figure 1 also shows how many additional rules had to be entered by the linguist in order for TA (Translator's Assistant—which combines LA and a document authoring and translation system) to adequately translate subsequent texts. As can be seen, the number of new rules required per text drops off dramatically after the introduction module has been completed. The great majority of rules that need to be written for subsequent texts are related to particular lexical items—not general linguistic phenomena. This demonstrates that the Grammar Introduction paradigm is effective in extracting rules needed in the translation process, which is in turn an indication of the coverage of the rules produced in the Grammar Introduction process.

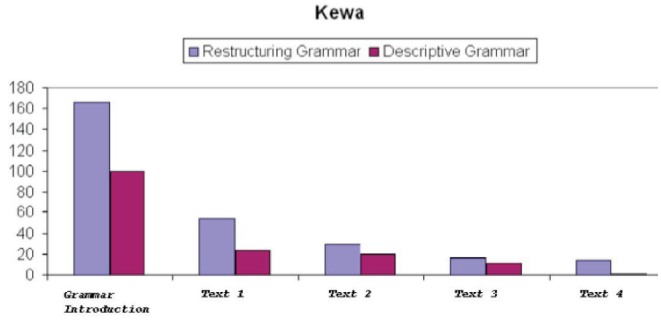

FIGURE 1: Utility of Grammar Introduction

3.3 ELICITATION QUESTIONNAIRE. Elicitation corpus[6] contains the current version of our meaning-based elicitation questionnaire. Here we go over the highlights. The sentences in the questionnaire in large part target the semantic features. The 'Features' video above presented the semantic features associated with objects, including person, number, identity

[5] See Allman 2010 for more details on this and other translation-oriented projects.

[6] A PDF of the elicitation corpus is stored in ScholarSpace, available here: http://scholarspace.manoa.hawaii.edu/bitstream/handle/10125/4500/elicitationcorpus.pdf

(definite entity or indefinite), referential distance (physical and contextual distance), and referential status (known/unknown). The first section of the questionnaire includes sentences that address each of these (refer to the questionnaire, where additional sentences are presented and additional explanations and context are given—these are meant to be examples only).

First mention:	A man hit the house.
First mention(pl):	Some men hit the house.
Subseq mention:	The man hit the house.
Physical distance:	
Near to speaker	This man hit the house.
Near listener	That man hit the house.
Contextual	A man hit the building. That man…
Number	2 men… The men…
	3 men… The men…
	4 men… The men…
	10 men… The men…

Likewise, the features associated with events, including time, aspect, mood, reflexivity, and polarity, all include elicitation sentences in the questionnaire (and so on for all the features).

The questionnaire also has sentences for different kinds of (semantic) argument frames/theta grids for events:

Intransitive:	The man slept.
Transitive:	The man hit the building.
Ditransitive:	The man gave the leaf to the woman.
PP arguments:	The man walked from the village to the store.
Clausal arguments:	
	The man promised to hit the building.
	The man told the woman that he hit the building.
	The man said, 'I hit the building.'
Adjunct arguments:	The man hit the building in the village.

We also have sections that exemplify different semantic relationships:

Kinship:	The mother of the man / Mary is the mother of the man.
Body parts:	The leg of the man / That is a leg.
Social roles:	The teacher of the boy / She is a teacher.

A section of the questionnaire is meant to uncover the pronoun paradigm.

Again, the current questionnaire reflects our progress at this specific point in time. As stated above, we intend to revisit its current content and integrate typology questionnaire techniques like BOAS. LA also allows the user to extend the included meaning-based corpus by adding naturally-occurring target text, a process that is beyond the scope of this

paper but is described in Beale & Allman (2011). Practically speaking, we have found LA—even in its current, imperfect state—to be of invaluable use in language description. In our document authoring and translation projects, in which LA is a component, there is an interplay between the work involved in documenting an extensive grammatical questionnaire and allowing the user to begin working through actual texts that are to be translated. We believe the give-and-take between grammar introduction and encoding knowledge for actual texts is beneficial. For one thing, the elicitation corpus has a limited vocabulary, which can artificially hide problems in the linguistic description. For this reason we include several pre-authored medical texts in LA so that users (and the final language description) can benefit from the experience and discipline of acquiring lexical and grammatical knowledge to cover actual texts.

4. ENCODING LEXICAL AND GRAMMATICAL KNOWLEDGE. This section will briefly describe the visual rule creation interface, explaining and exemplifying the various types of rules and for what purpose they are typically used. More importantly, we will attempt to draw out the methodology that we have found helpful in creating the target language's linguistic description of the elicitation corpus. Beale & Allman (2011) provides an example of how LA can be used to describe a specific linguistic phenomenon (direct/indirect possession in Maskelynes, Vanuatu) using naturally occurring texts.

Each sentence in the elicitation corpus has a semantic representation. The linguist needs to tell the computer how to realize each of the parts of that input semantic representation. This includes all the individual concepts, each of the semantic features (like tense, aspect, number, etc.) and all the relationships (such as case role relationships, adpositional relations, and discourse relations inherent in conjunctions). Backing up a bit, it is important once again to think about the overall nature of an LA project. All of these semantic representations describe the wide range of phenomena that we are interested in documenting. Once the linguist has created the lexical knowledge and grammatical rules so that LA's built-in text generator accurately translates the underlying semantics of these elicitation sentences, the language has been—to some extent—described. Section 3.2 described the further steps needed to move toward a complete description of the language.

Looking at the process in the order that the rules are actually applied by the computer text generation engine, from a Semantics-to-Target-Translation viewpoint, three steps need to be accomplished in order:

1. The lexicon specifies which target words, if any, translate the concepts in the semantic representation. Note: often there is not a one-to-one correspondence between concept and target words. This is resolved in the following steps.

2. Rules that convert the semantic representation into syntactic structures and features appropriate to the target language constitute the second step. As part of this step, all target language syntactic features must be identified and defined, including clause-, phrase-, and word-level features. An example of these target features is the syntactic function of noun phrases in English, e.g., Subject, Direct Object, etc. The semantic roles, such as AGENT and PATIENT, must be converted to these target-specific surface features (a process often dependent on the semantics of the event and/or the target language

verb inserted in step 1). Also part of this step is the process of converting the semantic representation into appropriate target syntactic structures. An example occurs in North Tanna (Vanuatu) in which the input semantic representation of [cl: *He took her to the house*] must be changed into something like [cl: *He took her* [cl: *they went to the house*]]. Another example occurs in Kewa (Papua New Guinea)[7] when the semantic representation of [cl: *He loves her*] is changed into [cl: *He sits happily with her*]. Note that the main function of these rules is to introduce syntactic structures (such as clauses, verbs, adpositions, adverbs, etc.) and target features, not to produce surface forms. For this reason we sometimes refer to the output of these rules as 'deep target structures' (as opposed to the 'surface' rules described next)[8] and to the rules themselves as 'structural adjustment rules.'

3. The final step in describing the target language realization of an input semantic description is to write the rules that will produce the actual surface output text from the 'deep' target language syntactic structures and features. These types of rules choose the correct forms of the target words and introduce affixes that realize the underlying deep syntactic structures and features. This is where the more 'usual' types of linguistic rules can be found, such as rules for agreement, morphological alternations, and phrase structure ordering rules.

Below we will refer to these steps as 1) lexicon, 2) deep syntax, and 3) surface output. The lexicon step stands alone somewhat. It can be carried out almost in isolation from the more grammatical steps (although the latter make use of the features and forms defined in the lexicon). The deep syntax and the surface output steps are quite inter-related in practice. In section 4.2 we will introduce the 'three main techniques for target grammar writing.' Those three techniques are aimed at implementing the deep syntax and surface output steps. There exists a potential for confusion between the three steps described above and the three techniques; therefore, we emphasize the differences here. The steps: 1) lexicon, 2) deep syntax, and 3) surface output are the three main steps that the text generation *engine* will go through. The three techniques described below refer to the *methodology* that we have found useful for the *linguist* to write the grammar rules. The machine follows the steps; the linguist should follow the methodology of section 4.2. First we describe the knowledge that is resident in the target lexicon and the tools that LA provides for acquiring it.

4.1 THE LA LEXICON.

▶ Play Demo in YouTube

VIDEO 5: A description of the lexicon, including the features and forms that can be created and used within LA.

[7] Kewa data is courtesy of Dr. Karl Franklin.

[8] The term 'deep' has several interpretations in the literature. We use it mainly to differentiate these kinds of rules from surface output rules, although various other aspects of its classical uses also apply.

The LA lexicon contains at least six different kinds of information:

- the root words and/or citation forms of a language
- a definition and gloss for each root
- mapping of root words to concepts that they (sometimes partially) realize
- defining features for root words
- defining forms for root words
- defining rules for the automatic creation of forms (for example, a rule specifies how to create the first person plural form hablamos in Spanish)

A seventh bit of information, to be discussed at the end of this document, is as follows:

- convenient links to all grammatical rules and all semantic representations in which the root word or the corresponding concept is present.

One of the main advantages of LA is that it is ideal for both fusional languages (which tend to combine more than one morpheme, or semantic unit, into an affix, with irregular forms more common) and agglutinative languages (which usually have more affixes, each of which encodes a single morpheme, with relatively fewer irregular forms).[9] And of course LA works well for isolating languages like Chinese or Thai in which there is little or no affixation.

The LA lexicon allows the linguist to define relevant features about individual root words that can be specified apart from their use in an actual sentence. These lexical features typically define classes of words which are relevant either in creating lexical forms or in grammatical rules. One example of this is the tendency for languages to group certain nouns together and treat a particular group differently than other groups in some way. English treats count nouns differently than non-count nouns. Spanish treats one group of nouns referred to as 'masculine' (which includes animals of the male gender along with many other nouns) differently than another group of nouns referred to as 'feminine.' Thai treats eggs differently than rocks. For verbs, the default transitivity of a verb is often an important input into its grammatical treatment. In languages that employ split ergativity, the expected behavior (nominative/accusative vs. ergative/absolutive) could be specified for each verb.

Lexical forms can also be specified for each root in the lexicon. Verbal inflection paradigms are usually best captured using forms in the lexicon. As will be shown below, the linguist can define rules that will automatically generate lexical forms.

As an example of a relatively fusional language we will examine (Latin-American) Spanish. For each verb we define a type feature in the lexicon (Figure 2) to specify whether the verb is an *-ar, -er,* or *-ir* verb.

[9] The comments about the relative number of irregular forms is a personal observation: cf. Turkish, Japanese, North Tanna for examples of agglutinative languages with few irregular forms.

	Stems	Glosses	type
1	aprend	learn	-er
2	habl	speak	-ar
3	ten	have	-er
4	viv	live	-ir

FIGURE 2: Features for Spanish Verbs

Note that the stem we use in the first column of Figure 2 removes this ending, and we define an infinitive form (see Figure 3). We could have alternately used the infinitive form in the 'stem' column since it is the more usual citation form and then defined a form that stripped off the infinitive ending; that form would then be the basis for the rules for the other forms.

Figure 3 (split into two for easier viewing) shows some of the forms for Spanish verbs. All of the cells that are not white were generated automatically by form rules (an example form rule is described below). When a new verb stem is added to the lexicon, a dialog window appears that allows the user to enter a definition, gloss, and values for each of the relevant features of the new stem. Then the program generates all the forms using the form rules. The user is then prompted to check the automatically generated forms and to make corrections when necessary. In this way we are able to handle fusional languages with irregular forms. The irregular forms of *tener* in Figure 3 are an example. Note that we are in the process of making form groups like these easier to manage. In Figure 3, we displayed the Present Indicative group. Currently for Spanish we must define 40 forms associated with the eight different tense/aspects (present indicative, subjunctive, etc.) each of which has five different subject-person/number combinations. In the near future, we would like to allow the user to define the eight forms for the tense/aspect, each of which could then visually expand to a table with the (user-definable) five different subject-person/number realizations. This will allow us to present conjugations and declensions in the more familiar format, as well as allowing for easier verification and modification for irregular forms.[10]

	Stems	Glosses	infinitive	present indic 1st sing
1	aprend	learn	aprender	aprendo
2	habl	speak	hablar	hablo
3	ten	have	tener	tengo
4	viv	live	vivir	vivo

present indic 2nd sing	present indic 3rd sing	present indic 1st pl	present indic 3rd pl
aprendes	aprende	aprendemos	aprenden
hablas	habla	hablamos	hablan
tienes	tiene	tenemos	tienen
vives	vive	vivimos	viven

FIGURE 3: Present Indicative Forms for Spanish Verbs

[10] The amount of information would remain the same (i.e., 40 combinations of tense/aspect with subject-person/number); it would simply be displayed in the more convenient format.

Figure 4 illustrates a rule that automatically generates the present indicative, first person plural form for Spanish. The three rows in Figure 4 reference the 'type' feature of the verb. For example, if a verb of type *-ar* is added to the lexicon, then a form will be generated that adds the *-amos* suffix to the stem. Similar rules generate the other forms. The more general capabilities of this kind of rule will be presented in section 4.3.

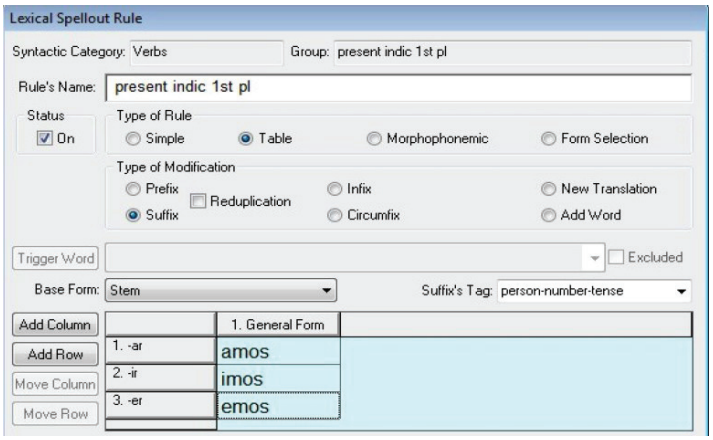

FIGURE 4: Lexical Form Generation Rule for Present Indicative, first person plural

How are these forms used in the grammatical part of the language description? We discuss this in more detail below, but suffice it to say for now that the linguist is able to write grammatical rules that will select any form for a given input sentence. For example, if the subject[11] of the input sentence is first person plural, and some other rule determines that Present Indicative should be used for the main verb in the input sentence, then there can be a surface output rule that will select the 'Present Indicative first person plural' form of the verb from among all the other possible forms of the verb present in the lexicon.

A relatively more agglutinative language will utilize fewer (often no) forms and, often, more features. Forms are usually contra-indicated in such languages because words in agglutinative languages are complex, comprised of a number of affixes attached to a stem. There can be a very large number of combinations of these affixes; typically a linguist would not want to enter each combination as a form. Furthermore, the rules for attaching affixes (for example, affix ordering rules, and in fact, the rules for selecting which affix realizes a given semantic concept or feature) are usually straightforward. And finally, our experience is that agglutinative languages tend to have fewer irregularities, and where they do exist, they are usually morphophonemic in nature, as opposed to lexically conditioned. All of this can be handled in general-purpose grammatical rules (to be described in section 4.3), as opposed to lexical rules that create forms directly associated with individual stems. Individual root words—for all languages, but particularly agglutinative ones—can be 'personalized' by defining features for them, for example, count and non-count, or transitive,

[11] The syntactic Subject will need to be identified by grammatical rules and assigned to a target feature created by the linguist.

intransitive, ditransitive, etc. General grammatical rules that generate the appropriate affixes can then reference the features of the stem to decide the correct affix.

For some languages it is possible to use either forms or general grammatical rules. Form creation rules have the equivalent functionality to the surface rules described in section 4.3 below—they are simply applied to create the forms shown in the lexicon. In Spanish, for example, we could have removed all of the forms and written general grammatical rules. The disadvantage of that for Spanish is that irregular forms would then have to be handled somehow. Forms allow irregularities associated with individual roots to be specified in the lexicon—probably where they belong. But if there were no (or few) irregular forms, then the linguist could simply define general grammatical rules to generate the needed affixes. This would have the advantage of making the lexical acquisition process easier, since the user would not be asked to verify all the forms each time a new lexeme is added. On the other hand, if general grammatical rules are used, exceptions need to be handled outside the lexicon, often requiring very specific rules.

The first step that our text generator takes when realizing an input semantic description is to associate target root words with concepts. We include an interface in the lexicon that enables the linguist to map concepts to the target roots that realize them. Note that the mapping might not be one- to-one; where some mismatch is present, structural adjustment rules (described in section 4.4) will be needed. Sometimes a concept will have no lexical mapping into the target language at all; again, structural adjustment rules will be used to describe how the concept is realized in the target language. Figure 5 shows a semantic representation of part of one of the meaning-based elicitation sentences (representing the English phrase *the man who John saw*) with target roots already associated with concepts (but before any grammatical rules were applied).

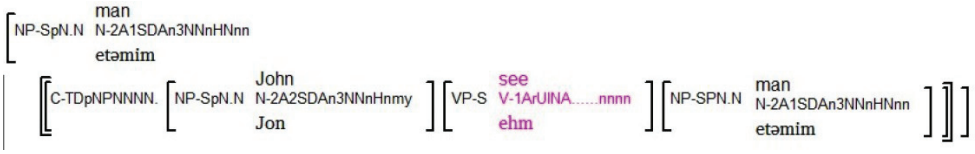

FIGURE 5: Semantic Representation with Target Stems Already Added

4.2 DEFINING TARGET LANGUAGE SYNTACTIC FEATURES. As described above, the lexicon defines relevant features about individual root words that can be specified apart from their use in an actual sentence. The linguist must also identify which non-lexical features the language employs in its grammar. These features can apply to the word, phrase, or clause level. An insightful analysis of these grammatical features is a key step toward an accurate and elegant description of a language. Practically speaking, they make writing the 'surface output' rules described in the next section relatively simple.

What are these features and how are they determined? A general rule of thumb follows: if there is some grammatical phenomenon in the language, and the linguist says something like, 'For a case like X, do this, and for Y, do that', then X and Y should probably be values of some target feature. Some examples:

• Certain adverbial clauses in English appear most often before the main clause; other adverbial clauses most often appear after the main clause.[12] So we can define a clause level feature called 'Adverbial Clause Position.' A 'deep syntactic' rule will then set this feature based on the actual concepts involved, and possibly based on other factors. The main point is that the surface output rule that performs the ordering of the adverbial clause will be simple—it will just reference the value of this feature.

• Languages often have complicated rules that determine which 'tense' and/or temporal or aspectual words and affixes to use for a given verb. The exact meaning of the word *tense* itself, or its range of values, often varies from language to language. Usually the best practice is to define a Tense feature (or one or more other features that are appropriate) and then use deep syntax rules to determine the correct value(s) based on the input semantics. This allows the surface output rule that produces the correct affix, word, or word order (or however the target language realizes temporal/aspectual semantics) to be simple, referring only to the value of the feature and then describing how it is realized.

• Substitute just about any target language affix, grammatically relevant word, or word order phenomenon for 'tense' in the paragraph above and the same suggestion applies. Some additional examples:

• Returning to the Spanish verbal suffixes, they encode a combination of the subject's person and number with the tense and mood of the action. There are several ways to approach this in practice. Perhaps the cleanest approach is to define a surface feature for the Spanish verb for each of these: namely, 'Subject Person,' 'Subject Number,' and 'Tense/Mood.' As will be shown below, a simple feature copying rule can be written that copies the person and number of the subject NOUN to the VERB and renames them 'Subject Person' and 'Subject Number.' More interesting will be the deep syntax rules that set the Tense/Mood feature of the target verb, as the conditions for determining it can be quite complicated. Once these target features are defined and set by the deep syntax rules, though, it is easy to construct a surface output rule that uses Subject Person, Subject Number, and Tense/Mood in a grammatical table rule that will be described below. (We will need to review the types of surface output rules, along with what constitutes an 'easy' rule, before the user will be able to concur with some of these statements.)

The recurring theme in all these examples is that we would like the surface output rules to be simple, optimally only referring to the value of a target feature (or features). In

[12] The English adverbial clause example highlights the tension between parsing and generation, and, in fact, brings out a possible weakness in our argument that LA comprehensively documents a language. LA is generation-oriented. The user only needs to describe the most likely realization of the semantics. All *possible* realization choices—which would be needed for robust parsing—are not needed. On the other hand, nothing prevents the linguist from entering multiple realizations.

that light, we present below the three main techniques for describing a particular linguistic phenomenon using LA:

1. Identify and define the necessary target feature(s), then
2. Write the surface rule that uses the feature(s), and finally
3. Write the deep structure rule(s) that will set the value of the feature(s) based on the input semantics.

In some cases, linguists might need help from experienced LA users to complete the deep syntax rules (technique three), but we have found that writing the surface output rules (technique two) is quite natural for linguists (in our admittedly limited experience). For this and other reasons, we suggest the order shown above. Also keep in mind that these three techniques presuppose that the user has already set up the lexically-related features and forms for each part of speech, which can usually be done apart from any thinking about grammatical rules.

An excellent question at this point would be, "How can the surface rules (in technique two) be tested without having the deep structure rules (in technique three) that set the target features used in the surface rules?" The answer lies in another important generalization: the linguist should work sequentially through each related set of elicitations, with testing and debugging performed after each. Each set in the elicitation corpus contains a single focus. For example, there is a set of sentences that address different verb times, another set that explores different aspectual information, etc. The linguist should take a set like this and develop the rules for the whole set, following the three techniques. Testing and debugging should occur after each relatively short development stage. Furthermore, the elicitation corpus is designed so that the initial sets contain sentences that are relatively simple. As the linguist progresses through the sets, new phenomena are introduced one at a time (in most cases) so that for each new set of sentences, only one previously unseen semantic phenomenon is introduced. Of course there will always be a bit of revising of earlier rules to be done as more examples are encountered, especially when—in the later stages of the project—naturally occurring texts are added to the elicitation corpus and pre-authored texts are translated.[13]

Practically speaking, how are the target features identified and added? In order to add a new target feature, the user simply clicks the 'Feature Set' button, navigates to the correct part of speech, adds the feature name, and then adds the set of possible values for that particular feature. On the other hand, actually identifying the correct set of target features to use is more complicated, since to do it well requires linguistic knowledge and intuition, knowledge of a particular target language, a bit of artistic ability, and some experience using LA. For this reason, we readily acknowledge that 1) we would not expect an untrained linguist to pick up LA and start writing efficient and accurate grammars right from the beginning, and 2) a training period is necessary in order to get 'the feel for' identifying target features that simplify surface rule creation. Beale & Allman (2011) describe the

[13] In order to avoid unintended changes, LA will store the results of translations; then, after changes to the grammar and/or lexicon have been made, the inputs can be re-translated. LA will mark any changes in red. Section 5 describes some of the other debugging functions of LA.

process for documenting alienable vs. inalienable possession in Maskelynes. A major focus of future work is to prepare additional training materials and to lead tutorials.

4.3 TARGET LANGUAGE RULES TO GENERATE SURFACE OUTPUT FORMS. In this section we identify and exemplify the different kinds of rules used to produce the surface output forms, i.e., the output translation.

We begin with the most familiar-looking rules. Figure 6 shows the interface provided for morphophonemic rules. The title bar of the window shown in Figure 6 says 'Spellout Rule'—our terminology for rules that directly produce surface output forms. The environment that must be matched for the rule to be applied can be described in terms of alphabetic characters or phonetic features. We have attempted to make the visual rule interface as natural and easy to use as possible. A large variety of morphological alternations, including reduplication, can be described.

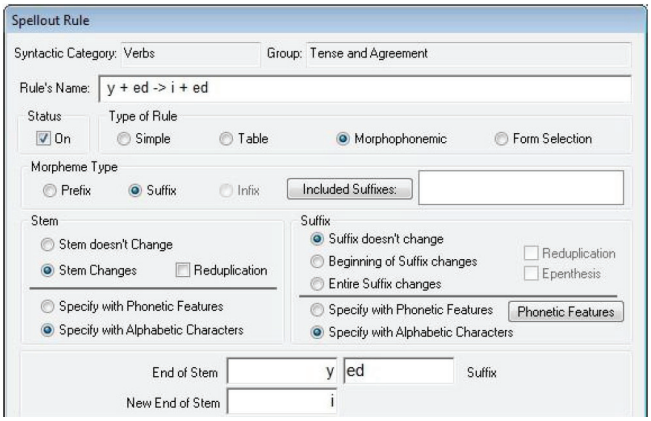

FIGURE 6: Morphophonemic Rules

▶ Play Demo in YouTube

VIDEO 6: Some examples of the visual rule creation interface for morphophonemic rules and their use.

Figure 7 shows an example of a simplified Phrase Structure Rule that orders the constituents of an English clause. Writing phrase structure rules is typically trivial once appropriate target features have been defined and set by earlier rules. In the example below, notice that there are multiple NPs, but each is clearly distinguished by a particular target feature value. To construct a phrase structure ordering rule, the visual interface allows you to add a constituent of any type by double-clicking on the head (the clause label in this example) and then choosing from among the possible constituent types defined for the language. The interface then presents the set of features associated with that constituent type. The user can select any required features via checkboxes. The program will insert the

specified constituent at the beginning of the list of constituents. The user can then drag the constituent to the correct spot in the list.

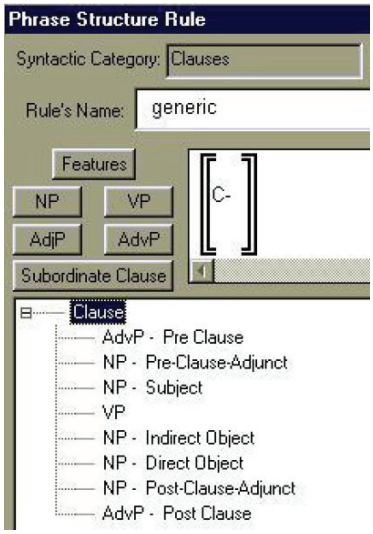

FIGURE 7: Phrase Structure Ordering Rule

▶ Play Demo in YouTube

VIDEO 7: Some examples of the visual rule creation interface for phrase structure ordering rules and their use.

Figure 8 is an example of a feature copying rule. This type of rule is useful when a constituent needs to 'know' the value of a feature of a different constituent. In English, for example, the Number (and Person—also accomplished in this rule, though not shown) of the Subject NP needs to be copied to the verb in order to facilitate subject-verb agreement. The feature copying visual interface allows you to set up the required constituents and add any required features to each. The box in Figure 8 (that is rather hard to see) labeled 'Noun Phrase Grammatical Relation = Subject' appeared when a mouse passed over the NP-S. Thus, for this rule only, the specified features of the NOUN in the NP marked as Subject will be copied to the verb. The user clicks on the 'Specify Source' button and then clicks on the constituent from which the feature(s) should be copied (in this case, the N). The user then clicks on the 'Specify Target' button (which replaces the 'Specify Source' button at the appropriate time) and then clicks on the constituent to which the feature(s) should be copied (in this case the V). The interface then allows the user to select the feature or features to be copied and allows the user to rename them as appropriate for the target constituent. This type of rule begins to show off the point-and-click convenience of the visual interface.

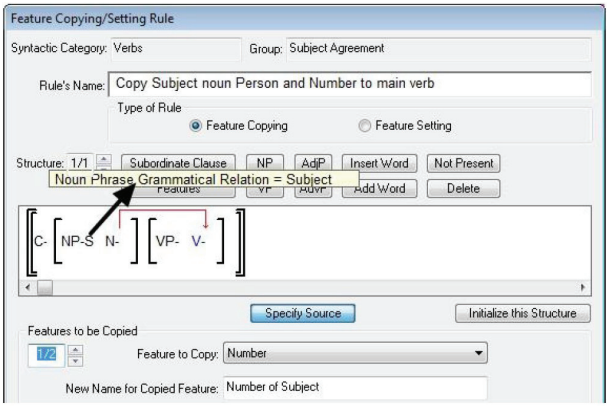

FIGURE 8: Feature Copying Rule

At the risk of belaboring the point, this English feature copying rule is another example of the 'three techniques' approach introduced in the previous section. We can write this rule without even knowing how the Subject NP will have its Noun Phrase Grammatical Relation feature set to Subject. We can just assume that feature exists and will be set by some other rule. We will worry about the rules that set the target features later (in section 4.4). It is important to note that a feature copying rule automatically creates the new target feature; it is not necessary to add it manually.

▶ Play Demo in YouTube

VIDEO 8: Some examples of the visual rule creation interface for feature copying rules and their use.

Figure 9 is an example of a table rule. The user is able to specify required features along the rows and columns (and globally required features in the top-corner cell). This example is a Noun table rule. At the appropriate time (all rules in LA are applied to the input sequentially), this rule will be applied to each noun in the input. In the rows and columns of the table, features can be specified that apply to the word itself, its immediate phrase, or its immediate clause. If the features of an input noun match the features of a row and column in the table, the contents of that cell will then be used either to add affixes to the word, add additional words (with user-defined POS), or change the current word—depending on which radio button the user clicks in the 'Type of Modification' section of the rule. In this example, the noun is converted into a personal pronoun. The 'Structures' button can be used to specify an arbitrarily complex input structure that must be matched for the rule to be applied. The 'Trigger Word' button can be used to restrict the rule to applying (or not applying) to a specific subset of target words.

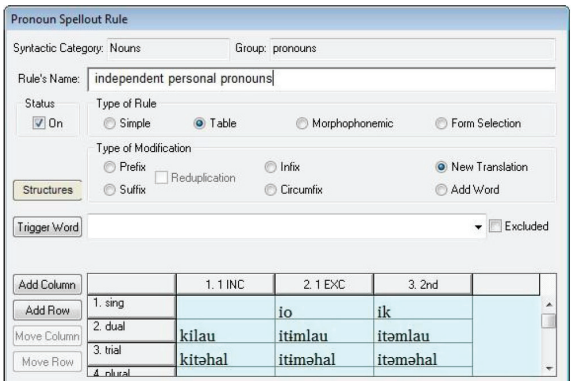

FIGURE 9: Table 'Spellout' Rule

Figure 10 is an example of a surface 'spellout' rule that appends a prefix. The features that are required for the rule to apply are entered by clicking the 'Features' button, at which time LA will present the list of features and their possible values that are defined for that constituent type (Verb, in this case). The user selects the appropriate feature(s), which are then displayed (in this case, 'Verb Inflection Type = nominalized'). This example is for a verb. Again, at the appropriate time, this rule will be applied to each verb in the input. If the input verb matches the features that are specified in the rule, as well as the required Structures and Trigger Words, the modification will take place: the addition of an affix, the addition of a modifying word (with user-defined POS), or a new translation for the input root. In this case a prefix with tag 'nominalized' is added to the input verb.

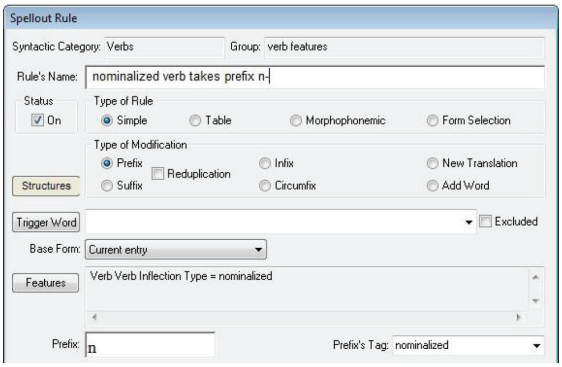

FIGURE 10: Affix 'Spellout' Rule

This concludes the overview of the major types of Surface Output rules available in LA. There are, in fact, other rules available, and as alluded to above, even the rules we did describe allow for several more advanced features we will not discuss here. Our desire, however, is that the relative simplicity of these rules is evident. We would argue that most linguists would be familiar with their function and would be able to adopt them without difficulty.

▶ Play Demo in YouTube

VIDEO 9: Some examples of the visual rule creation interface for surface output rules and their use.

4.4 STRUCTURAL ADJUSTMENT RULES TO CHANGE SEMANTIC CONCEPTS, SEMANTIC FEATURES, AND SEMANTIC CONSTRUCTIONS INTO TARGET LANGUAGE DEEP SYNTACTIC STRUCTURES AND FEATURES. This section, which describes how to write rules to describe the deep structure of a language, tends to be a bit more difficult for some people. The proper use of these rules necessitates an understanding of the input semantics, learning which types of rules can be used to accomplish which goals, and a clear understanding of the target language. For this reason we often suggest that—at least initially—a linguist work with someone who is experienced in LA structural adjustment rules.

In this step of language description, rules need to be written that convert certain semantic representations to 'deep' syntactic forms that are appropriate to the target language (again, we use the term 'deep' mainly to distinguish these from the 'surface' rules of the previous section). These rules perform the following functions:

- They introduce target language syntactic features that result from a distillation of semantic inputs into one or more target features that can then easily be used in Surface Output rules.

- They introduce new target language structures and/or words that realize some part of the semantics but are not in themselves direct translations of a concept (for example, adding target language adpositions to realize case roles).

- They deal with concept-level mismatch in semantic class (for example, a semantic property not being realized as surface adjective) or ontological mismatch (for example, a TEACHER concept realized as 'person who brings up thinking').

We explore each of these below. We characterize these types of rules as 'structural adjustment rules.' It is good to keep in mind that a single structural adjustment rule can accomplish all three of the functions listed above: for example, a single structural adjustment rule that is triggered by a specific input event could introduce a target feature, add a target language preposition to realize a case role, and expand the input event concept into some sort of complex target language phrase. Thus the three categories are more pedagogical than practical.

Returning to the discussion of the 'three steps' at the beginning of section 4, it is important to understand that the structural adjustment (or 'deep syntax') rules are processed before the surface output rules. In computer science terms, the structural adjustment rules can 'feed' the surface rules; that is, they can set up the structures that are expected inputs in the later rules. In fact, all rules in an LA grammar are executed sequentially. Thus a sequentially earlier structural adjustment rule can feed a later one. Of course an earlier rule can also unexpectedly *interfere* with a later rule, leading to 'bugs.' An especially onerous in-

stance of this occurs when the linguist adds a rule that 'destructively interferes' with a rule that he or she added days or weeks before. Section 5 discusses the debugging facilities that are included with LA, but the most important programming technique we urge linguists to take advantage of is regularly checking that previously translated texts do not change after rules are added or modified. The linguist will be working through the meaning-based elicitation corpus, and later in the process adding naturally occurring texts as well as translating the pre-authored texts that come with LA. At any point in this process, the user can have LA save the translations of all inputs that have been 'handled.' Later—preferably at regular intervals whenever a significant amount of change has occurred in the grammar—the user can ask LA to compare the current translations to the saved translations. LA will print out a side-by-side comparison, with any differences highlighted in red. If there are differences, the user can take advantage of LA's debugging facilities, described in section 5, to identify the rule(s) responsible for them.

4.4.1 DEEP SYNTAX RULES THAT INTRODUCE TARGET FEATURES. We begin with the function of Deep Syntax rules that is in some ways the most critical for the user to understand. We have already argued several times that well-thought-out target features render the Surface Output rules relatively simple. It is therefore extremely important to know the kinds of resources available for setting these target features. We have, in fact, already seen one method for setting a target feature. Figure 8 above displayed a 'feature copying rule'. Feature copying rules allow the user to copy a feature from one constituent to another constituent of a different type. As part of this process, the user gives the newly copied feature a name that is appropriate to the new constituent type. Thus, this is an example of creating a new target language feature. It might, in fact, be beneficial from a pedagogical point of view to move feature copying rules into the category of Deep Syntax rules. We have resisted such a move because the feature copying rules tend to be familiar to linguists, along with the other Surface Output rules, and we therefore leave them together. But it is helpful to realize that they have a similar function to the structural adjustment rule that we are about to describe. Rules of the following sort are simply more complex feature copying rules.

In English, we might want to specify that a certain class of clause-modifying prepositional phrases (like 'in the morning') attach to the front of the clause. Figure 11 is an example structural adjustment rule that sets the target feature, Noun Phrase Surface Type, to the Pre-Clause-Adjunct value in the presence of the IN concept (sense E, a temporal sense of IN). There might be various other types of NPs that we would also want to attach as a Pre-Clause-Adjunct. We could, for example, add rules similar to this one that set Noun Phrase Surface Type to Pre-Clause-Adjunct for NPs that contain a VP or some other kind of constituent, or for all NPs that have a particular semantic feature, or for all NPs that contain a specific concept or target word, etc. Obviously, there will be other rules that set Noun Phrase Surface Type to other values in other situations. The point is that we can write rules of this type for all of those, setting this particular feature to its proper value. The surface rule that interprets or 'implements' the intention of this feature can then be simple. In Figure 7 we saw a phrase structure ordering rule for clauses. Near the top of the ordering is 'NP - Pre-Clause-Adjunct,' a shorthand identifier for 'Noun Phrase Surface Type = Pre-Clause-Adjunct.' This is completely typical of how the linguist should go about realizing phrase structure ordering for the target language: write a surface realization rule that uses

the value of a target feature, and then write a rule or series of rules such as the one in Figure 11 that sets that feature correctly.

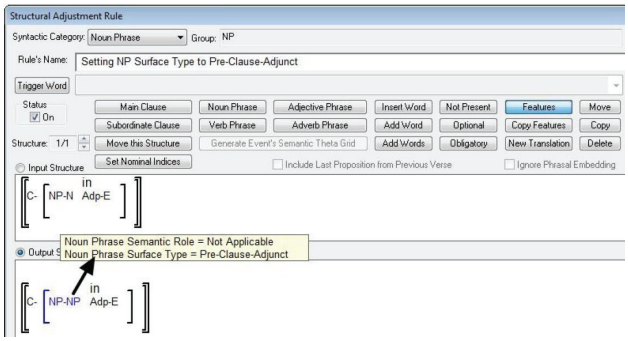

FIGURE 11: Structural Adjustment Rule that Sets Target Feature

Each structural adjustment rule has an input zone and an output zone. In order for the rule to be triggered by a particular input, all of the clauses, phrases, concepts, and features in the input zone of the rule must match a section of the input semantic representation. LA contains a convenient visual interface for constructing structural adjustment rules (and, in fact, it contains a facility for copy-pasting a portion of an input sentence's semantic representation directly into the input of the rule). The buttons at the top of the rule are used to insert constituents and concepts, and are also used to mark certain elements as optional, obligatory, or not present. The 'features' button is used to add any semantic or target feature to either the input zone or the output zone. Once the input structure is specified, the output structure radio button should be clicked, at which time the input structures are automatically transferred to the output.[14] Then the user can simply make the desired changes to the output. To move a constituent to a different phrase, the user clicks the 'Move' button, clicks on the constituent to be moved, then clicks the phrase to which it should move. A visual trace of the operation is left in the rule. The 'Copy' button works in a similar manner. The 'Delete' button is used to delete constituents. Additional constituents, concepts, and target words can be added to the output zone using the appropriate buttons.

▶ Play Demo in YouTube

VIDEO 10: Some examples of the visual rule creation interface for structural adjustment rules and their use.

In the example above, the input structure is a clause with an embedded NP on which the user has set the 'N' feature, short for 'Noun Phrase Semantic Role = Not Applicable.' That is the input semantic feature that we use in our English grammar for adjunct NPs. The

[14] The user can go back later and adjust the input, which will automatically adjust any existing output structure when it is possible to do so without ambiguity.

IN concept (sense E, the temporal sense) is then added to the NP.[15] IN does not have to be the only concept this rule can match on. Multiple concepts can be added for any word in the input, or concepts can be left completely underspecified (for example, a rule can match on any verb/event). The output of this rule is the same as the input, except that the P feature was added to the NP, short for 'Noun Phrase Surface Type = Pre-Clause-Adjunct.' Structural adjustment rules can be arbitrarily complex, but are often very simple. For this subsection, the purpose of the rule is to set a target feature (for use in the Surface Output rules) on the basis of a (sometimes complex) analysis of the input semantics.

4.4.2 DEEP SYNTAX RULES THAT INTRODUCE NEW TARGET LANGUAGE STRUCTURES AND WORDS NOT DIRECTLY INVOLVED IN TRANSLATING CONCEPTS. We have already seen, in section 4.4.1, all of the functionality of structural adjustment rules. All that remains is to describe how else they can be used. Sections 4.4.2 and 4.4.3 describe the other two main uses.

Semantic events often require a type of structural adjustment rule that we have termed 'theta grid adjustment' rules. As the name implies, they are primarily used for implementing the theta grid, or case frames, of a verb. For example, prepositions are sometimes needed to render certain case roles in English. The English rule shown in Figure 12 specifies an input clause with a MOVE event (this happens to be sense 'B' of MOVE—the 'change residence' sense). There are also three NPs in the input semantics with the semantic roles 'participant/agent,' 'source,' and 'destination' (specified by the 'p,' 's' and 'd' letters connected to the NPs in the input; the actual names and values of the features are displayed when the user rests the cursor over an NP symbol). The corresponding output is then specified. In this case, the participant becomes a subject, the source becomes a 'Post-Clause-Adjunct' with a 'from' preposition, and the destination is also a 'Post-Clause-Adjunct' with a 'to' preposition.[16] The important point for this discussion is the insertion of the target prepositions to aid in realizing the source and destination case roles. Since event frames so often require some sort of structural adjustment in order to be realized in a target language, we have supplied these theta grid adjustment rules for each sense of each event in the ontology, pre-filling their inputs with the constituents that are applicable for each sense. Theta grid adjustment rules are basically just a convenience; they are simply built-in structural adjustment rules that are ready to use for each event in the ontology. They are also often used when an event is involved in a semantic mismatch, as described in the next section.

[15] For various reasons beyond the scope of this paper, we do not create a PP syntactic phrase, but simply add adpositions at the NP level.

[16] Our English grammar uses the word 'Adjunct' to mean any NP that is not in a subject or object position, i.e., any NP that requires a preposition.

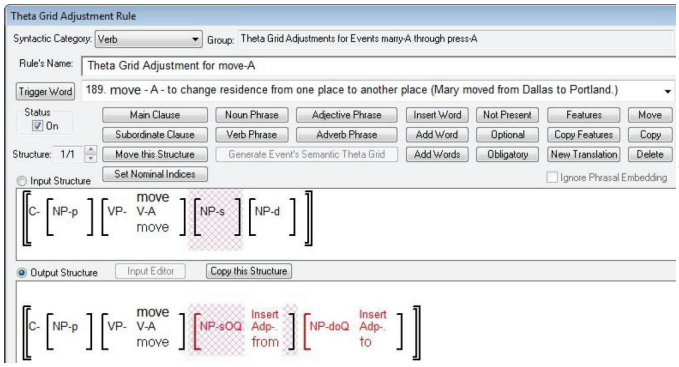

Figure 12: Deep Syntax Rule that Adds Prepositions to Realize Case Roles

▶ Play Demo in YouTube

VIDEO 11: Some examples of the visual rule creation interface for theta-grid adjustment rules and their use.

4.4.3 DEEP SYNTAX RULES THAT DEAL WITH SEMANTIC MISMATCH AT THE CONCEPTUAL LEVEL. We have left the most difficult case for last. The best way to introduce the need for these kinds of rules is through some examples. Figure 13 contains a list of examples of semantic mismatch in Kewa, a language group in Papua New Guinea.

Kewa* Structural Adjustment Rule Examples

	Semantic Representation	Kewa Equivalent
Ruth 2:1	X respects Y The people respected Boaz.	X lifts up Y's name The people lifted up Boaz's name. Onáá-mé Boasi-ná bi minasaa-simi. people-SUBJ Boaz-POSS name lift.up-REMPAST
Ruth 2:16	X be kind to Y Boaz said, "Be kind to Ruth."	X puts good thoughts toward Y Boaz said, "Put good thoughts toward Ruth." Boasi-mi ágaa gupa lakesa: Ruti madaa épé kóne sá-lepa sa. Boaz-SUBJ talk like.this tell.3Sg.REMPAST: Ruth toward good thought put-IMPTENSE say.3Sg.PAST
Ruth 4:15	X loves Y Ruth loves you and she	X sits happily with Y Ruth sits happily with you and she ... Ruti-mi ne rana plr-u-a ... Ruth-SUBJ 2SG happily sit-HAB-SS ...
Ruth 3:5	X obeys Y Ruth said to Naomi, "I will obey you."	X hears Y's talk Ruth said to Naomi, "I will hear your talk." Ruti-mi Naomi ágaa gupa sa: né-mé ne-ná ágaa paga-lua sa. Ruth-SUBJ Naomi talk like.this say.3Sg.REMPAST: 1Sg-SUBJ 2Sg-POSS talk hear-FUT say.3Sg.PAST
Ruth 1:14	X goodbyes Y Orpah said goodbye to Naomi.	X says to Y, "Go to sleep." Orpah said to Naomi, "Go to sleep." Opa-mé ágaa gupa sa: pépta-pe sa. Orpah-SUBJ talk like.this say.3Sg.REMPAST: sleep-IMPTENSE say.3Sg.PAST
Discourse 1:5	X greets Y David greeted his brothers.	X says to Y, "Where are you going?" David said to his brothers, "Where are you going?" Tepiti-mi nipú-ná áme-nu ágaa sa: nimi-mi aa-para púlumi pae sa? David-SUBJ 3Sg-POSS brother-Pl talk say.3Sg.REMPAST: 2Pl-SUBJ which-place go.3Pl.PRES INT say.3Sg.PAST

*All Kewa data courtesy of Dr. Karl Franklin.

FIGURE 13: Structural Adjustment Rule Examples from Kewa

These examples are fairly self-explanatory: in each case there is no one-to-one correspondence between the main event concept and the target language. Each of these examples is readily addressed with a structural adjustment rule.

In order to avoid giving the impression that everything in LA is very simple, we include in Figure 14 the rule necessary to realize GREET in Kewa. Figure 14 is difficult to read, so we include Figure 15—an enlargement of a portion of the rule in which the original Patient NOUN is copied to the subordinate clause and its Person feature is set to 'second

person.' The copied NOUN will retain the same Number feature as the original. A few other minor adjustments were made in Figure 14, but overall it should be clear how the rule works. Even for a relatively complex example, the rule is not terribly hard to understand. After a tutorial and some practice, linguists should be able to create these rules on their own—and even enjoy the experience!

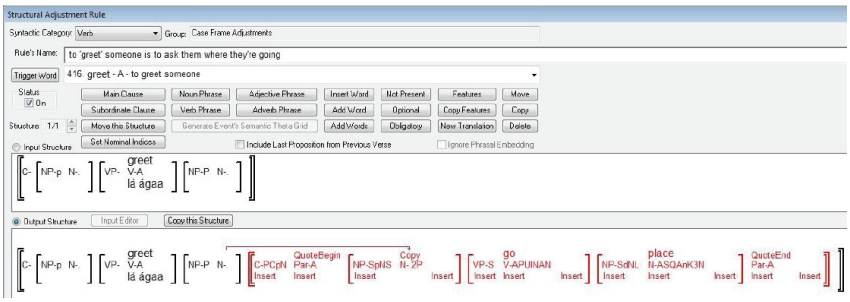

FIGURE 14: Structural Adjustment Rule Example: GREET in Kewa

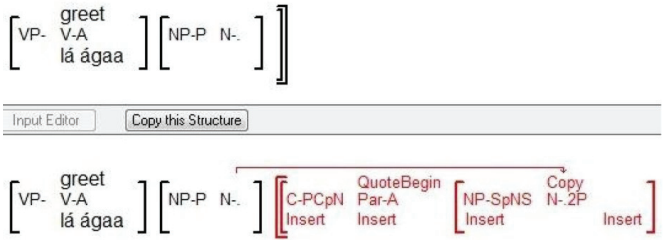

FIGURE 15: Enlargement Showing Copying of Noun to Subordinate Clause

Before leaving this section, we want to present an example of semantic class mismatch. A classic example is a semantic property being realized as a target verb. Figure 16 is another Kewa example in which the translation of 'X is PROUD of Y' would have the English gloss 'X puts good thoughts Y.'

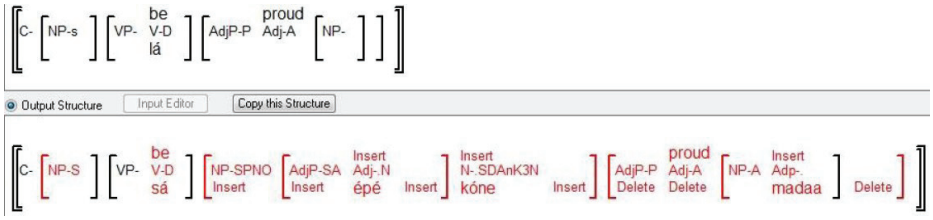

FIGURE 16: Conceptual Class Mismatch: ADJECTIVE as a VERB

The PROUD property and the ADJP boundaries are deleted, the generic translation of this sense of BE is changed to *sa'* ('put'), the translations for 'good thoughts' are added, and various adjustments are made to the target NP-level features to ensure they are ordered correctly in the surface output.

4.5 OUTPUT OF LA. As discussed at the beginning of section 4 and again in section 4.4, LA processes the grammatical rules sequentially, starting with the structural adjustment (or deep structure) rules and then moving to the surface output rules. The rules within each of these two types of rules are also processed sequentially. LA is a *deterministic* processor; that is, each rule is applied if its input conditions match, producing a single output that is then used as the input to the next rule. Multiple possible structures are never created. The surface output rules produce the output strings of the language and order them correctly; thus the final output of LA is a single target language string that is a translation of the input semantics.

5. ANCILLARY FUNCTIONS OF LA. LA contains many features aimed at making it more useful and easier to use. The most important of these is the ability to generate output grammars in text or HTML. We continue to work on improving this feature in collaboration with our users. LA also has several features meant to aid the user in troubleshooting, as demonstrated in the following video.

▶ Play Demo in YouTube

VIDEO 12: An overview of the ancillary functions of LA used in debugging and creating output language descriptions

6. CONCLUSION AND DISCUSSION. We have presented an overview of Linguist's Assistant. Our purpose was to introduce LA to field linguists, giving them enough information and examples to understand the nature and scope of LA. We encourage the reader to view the demonstration videos included in this paper, visit our website at http://ilit.umbc.edu/sbeale/LA, or contact the author at sbeale@cs.umbc.edu for more information. The LA program can be downloaded from the website and used for non-commercial purposes. We have also included tutorials, papers, several older tutorial movies, and links to several conference papers that describe LA and our related translation applications.

The most important question to address in conclusion is whether LA is an appropriate tool for language documentation. A criticism levied against LA is that modern field linguists prefer to use naturally-occurring texts as the basis for their language analysis and documentation as opposed to LA's meaning-based elicitation process. Having read a draft of this paper, a respected linguist offered some criticism: 'I am, in general, a bit reluctant to use ready-made questionnaires, for all sorts of reasons—some of which you mention yourself. It so happens that my personal interest has always been on naturalistic speech.… I have always paid a lot of attention to what actually shows up in everyday spoken speech, as opposed to what could exist 'grammatically' but is never heard. I've always wondered why so many grammars or articles in linguistics work on sentences such as "*The man sees the woman*." which don't appear ever in naturalistic speech' (Alex François, personal communication).

We believe that a middle ground is not only possible but profitable. Ameka et al. (2006:11) state that 'limiting what the grammar should account for to a corpus [of naturally occurring texts] also overlooks the fact that speakers may have quite clear and reveal-

ing judgements' and 'the view…that grammars should be answerable just to a published corpus seems an extreme position in practical terms.' Gippert et al. (2006:4) warn that 'without theoretical grounding language documentation is in the danger of producing 'data graveyards', i.e. large heaps of data with little or no use to anyone.' As was stated in the introduction, we believe that the combination of semantically-motivated and textually-motivated description provides an ideal balance. The meaning-based elicitation is general and uniform across languages. It provides an efficient and relatively comprehensive standard for describing the majority of the linguistic phenomena in a language. We have used this approach to document five languages and to produce a significant amount of high-quality translations, which themselves are a good argument for the validity of the underlying language description. Beale et al. (2005) and Allman & Beale (2004, 2006) contain documentation on the evaluations of the translations produced. Of course, a side benefit of LA is that the computational language descriptions can be used in translation applications.

The efficiency gained by using LA is another important benefit. Practically speaking, languages around the world are dying at an alarming rate. This fact is the primary reason that language documentation is experiencing a resurgence. It might be theoretically ideal to send a linguist into each endangered language community to gather a sufficient amount of naturally occurring texts in order to perform a detailed analysis and description of the language.[17] But practically speaking, the urgency of our current situation demands that we consider other, more efficient alternatives—at least in some cases. To back up this claim, we point to the Summer Institute of Linguistics, which has officially adopted a 'Vision 2025' goal, which proposes that work should be started in every language where it is needed by 2025. A strong argument could be made that 2025 is too late—many languages will already be extinct by then. Gippert et al. (2006) state that 'the task of compiling a language documentation is enormous, and there is no principled upper limit for it... every specific documentation project will have to limit its scope and set specific targets.'

Therefore, LA's meaning-based elicitation corpus and associated grammatical description methodology *could* be seen as an alternative to a comprehensive analysis of naturally occurring texts. LA could form the basis of a structured and efficient language documentation project. But is it enough? In some cases, it may have to be. However, where time and resources permit, LA encourages the addition of naturally occurring texts to its standard elicitation corpus. Beale & Allman (2011) describe using such an approach to document direct and indirect possession in Maskelynes. Furthermore, a possible (though currently undocumented) mode of work in LA is to replace or adjust the meaning-based elicitation corpus that comes with LA with naturally occurring texts. This method would involve finding naturally occurring texts that would cover the topics in the elicitation corpus, and then adding those texts (after semantically analyzing them) to it.

And finally, although "*The man started hitting the building*" might not translate into a naturally occurring sentence in a particular language, the semantic feature of *inception* probably does. The LA elicitation corpus groups together sets of related phenomena, making it clear what is being focused on in each group. The instructions emphasize that the linguist and/or native speaker can freely substitute different words or concepts in the pe-

[17] However, Ameka et al. (2006:15) state that meaning-based grammars (as opposed to this form-based methodology) "should remain an important descriptive goal."

riphery of any sentence. Furthermore, the elicitation corpus is meaning-based. So theoretically speaking, the concept of a human male beginning to strike a building *should* be communicable in any language. The instructions make clear that if any requested elicitation is difficult to encode in the language, then the linguist or native speaker should simply skip or adjust that input.

The above arguments show that using LA's meaning-based elicitation methodology might be a practical alternative to traditional, but less efficient, research paradigms that emphasize naturally occurring texts. And, of course, LA can be used within a more traditional language documentation project as another tool in the descriptive arsenal. The issues are complex and these responses are not conclusive, but we hope that they will be a basis for further discussion and debate.

References

Allman, Tod. 2010. The Translator's assistant: a multilingual natural language generator based on linguistic universals, typologies, and primitives. Arlington, TX: University of Texas dissertation.

Allman, Tod & Stephen Beale. 2006. A natural language generator for minority languages. In Proceedings of Speech and Language /technology for Minority Languages (SALTMIL). Genoa, Italy.

Allman, Tod & Stephen Beale. 2004. An environment for quick ramp-up multi-lingual authoring. *International Journal of Translation*, Vol 16, No. 1.

Ameka, Felix, Alan Dench & Nicholas Evans. 2006. *Catching language: the standard challenge of grammar writing*. Berlin: Mouton de Gruyter.

Bateman, John. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering 3*(1), 15-55.

Beale, Stephen & Tod Allman. 2011. Linguist's Assistant: a resource for linguists. In Proceedings of the Fifth International Joint Conference on Natural Language Processing (IJCNLP-11), The 9th Workshop on Asian Language Resources, Chiang Mai, Thailand.

Beale, Stephen, Sergei Nirenburg, Marjorie McShane & Tod Allman. 2005. Document authoring the Bible for minority language translation. In Proceedings of MT-Summit. Phuket, Thailand.

Beale, Stephen, Sergei Nirenberg and Kavi Mahesh. 1995. Semantic analysis in the Microcosmos machine translation project. In Proceedings of Symposium on Natural Language Processing, Kaset Sart University, Bangkok, Thailand.

Bender, Emily, Scott Drellishak, Antske Fokkens, Michael Wayne Goodman Daniel P. Mills, Laurie Poulson & Safiyyah Saleem. 2010. Grammar prototyping and testing with the LinGO Grammar Matrix Customization System. *Proceedings of the ACL 2010 System Demonstrations*.

Black, Sheryl & Andrew Black. 2009. PAWS: Parser and writer for syntax: drafting syntactic grammars in the third wave. http://www.sil.org/silepubs/Pubs/51432/SILForum2009-002.pdf

Bouquiaux, Luc & Jacqueline Thomas. *Studying and describing unwritten languages*. Dallas: SIL International.

Comrie, Bernard & Norval Smith. 1977. *Lingua Descriptive Studies: Questionnaire*. Amsterdam: North-Holland. 72 pp.
Gippert, Jost, Nikolaus Himmelmann & Ulrike Mosel. 2006. *Essentials of language documentation*. Berlin: Mouton de Gruyter.
Givón , Talmy. 1984. *Syntax: a functional-typological introduction*. Amsterdam: John Benjamins Publishing Company.
McShane, Marjorie, Sergei Nirenburg, Jim Cowie & Ron Zacharski. 2002. Embedding knowledge elicitation and MT systems within a single architecture. Machine Translation 17(4):271-305.
Nirenburg, Sergei, & Victor Raskin. 2004. *Ontological Semantics*. MIT Press, Cambridge, MA.
Nyberg III, Eric H. and Teruko Mitamura. 1992. 'The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains' *Proceedings of COLING-9*.
Payne, Thomas. 1997. *Describing morpho-syntax: a guide for field linguists*. Cambridge: Cambridge University Press.
Probst, Katharina, Lori Levin, Erik Petersen, Alon Lavie & Jamie Carbonell. 2003. 'MT for minority languages using elicitation-based learning of syntactic transfer rules.' In Machine Translation 17(4): 245-270.

Stephen Beale
sbeale@cs.umbc.edu